(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 December 2003 (11.12.2003)

PCT

(10) International Publication Number
WO 03/102874 A1

(51) International Patent Classification[7]: G06T 11/20

(21) International Application Number: PCT/US03/17088

(22) International Filing Date: 29 May 2003 (29.05.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/159,989 31 May 2002 (31.05.2002) US

(71) Applicant: ENVIVIO, INC. [US/US]; 801 Gateway Boulevard, Suite 402, South San Francisco, CA 94080 (US).

(72) Inventors: DENIAU, Eric; La Prise, F-35190 La Baussaine (FR). CAZOULAT, Renaud; 60, rue pollain Duparc, F-35000 Rennes (US).

(74) Agent: SWERNOFSKY, Steven, A.; Swernofsky Law Group_PC, P.O. Box 390013, Moutain View, CA 94039-0013 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

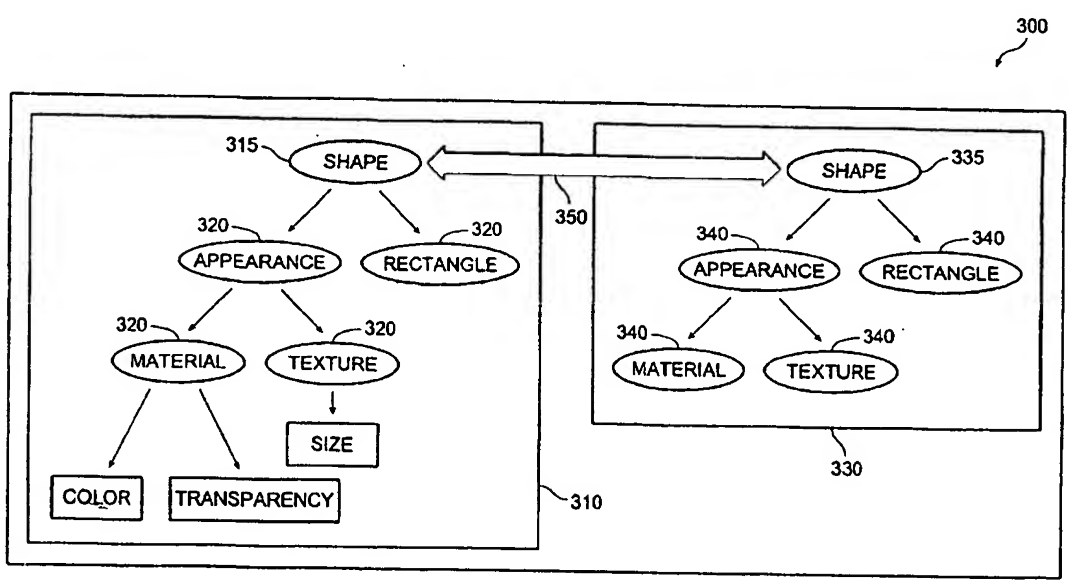
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*

Published:
— *with international search report*

[Continued on next page]

(54) Title: OPTIMIZED MIXED MEDIA RENDERING



(57) Abstract: A method (400) and system (100) for rendering mixed media at a presentation workstation (120) in response to a mixed media stream (140), in which the amount of computation is minimized. A tree structure (310, 330) includes nodes (315, 335), each of which might point to sub-trees (320, 340), and each of which designates a geometric object, a transformation to be applied to a geometric object, or a time-varying presentation. A scene (300) includes nodes for each element in the scene. Depending from these would be nodes describing aspects of those elements, such as their texture and shape. A root node (315, 335) joins the descriptions of each of the objects in the scene. Optimizing the amount of computation includes selecting nodes in the tree and calculating how

WO 03/102874 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

OPTIMIZED MIXED MEDIA RENDERING

Background of the Invention

1. *Field of the Invention*

The invention relates to mixed media rendering.

2. *Related Art*

It is often desirable to send more than one type of media (known as "mixed media") from a content server (or "originating server") to a client device (or "terminal") for presentation to a user. Mixed media can include: video, still images, vector graphics, bit maps, sounds, and other information for presentation to the user. MPEG-4 is one known method for encoding such mixed media scenes. MPEG-4 has a graph data structure, including the nodes of the graph, and links or relationships between pairs of those nodes. Information in this graph data structure describes the constituent elements of the scene to be presented.

One known problem with using MPEG-4 is that graphical display of different content and different media types (collectively known as objects) can raise significant performance issues. For example, when two objects or parts of objects occupy approximately the same spatial location in a MPEG-4 presentation, it may be unclear which object is to be displayed to a viewer. Similar problems exist when one of the objects is transparent or translucent (thus requiring that some portion of the both images must still be shown). An MPEG-4 display may include a large number of such objects, each requiring significant computation for correct presentation.

Another known problem is that the frame rate (that is, the number of frames that are displayed per second) is inversely proportional to the complexity of the graphical display. It is difficult to maintain a sufficiently fast frame rate when computing a complex stream without compromising the quality of the display.

Summary of the Invention

The invention provides a method and system for rendering mixed media at a terminal in response to a mixed media stream (such as an MPEG-4 stream) in which the amount of computation is minimized while the entire mixed media presentation is accurately rendered and presented.

In one aspect of the invention, two different tree structures are used to represent an object in a scene. The first of these tree structures is called a scene graph. The scene graph includes a root node and a set of dependent nodes. The root node includes a geometric shape, such as a rectangle, that corresponds to an object in the presentation. The dependent nodes can designate various characteristics of the object such as color, transparency, texture, rotation, volume, size and other similar features. These nodes provide placeholders for values associated with these various characteristics and shapes. A second tree structure, called a presenter graph is related to the first tree structure. Similar to the scene graph, the presenter graph includes a root node and a set of dependent nodes. As the presenter graph is traversed, the instructions for rendering an object in a scene are obtained using the values included in the scene graph. There is a one-to-one correspondence between scene graphs and presenter graphs. Scene graphs include information such as the static values that are used to describe the elements of each scene. Presenter graphs include instructions for rendering dynamic changes in scenes efficiently using values already included in (or computed from) those scene graphs.

In another aspect of the invention, a join node couples the various tree structures and provides information about how the objects are integrated into a single scene. Subtrees can also depend from the nodes in a presenter graph and from the nodes of a scene graph. Multiple scene graphs and multiple presenter graphs can be used in the rendering of an entire multi-media presentation.

In a second aspect of the invention, a processor at the terminal selects nodes in the tree and calculates whether to render each node in the tree and if so, how to render that node. Similar calculations are performed for subtrees that may depend from each node.

When possible, the processor calculates rendering information for only those portions of objects that will ultimately be shown to the user.

This technique results in significant savings in computational resources used to encode and render an object. For example, this technique results in a substantial savings of resources when encoding a presentation that includes several small objects moving on top of large static objects (such as birds flying across an unchanging sky) because it is only necessary to re-render the changing portions of the drawings. In a second example, there is also large saving of resources when zooming in or enlarging a single object. In this second example, it is not necessary to redraw all of the objects that are hidden behind the enlarged object. Similar savings can also be obtained when determining how to render sound such that it may not be necessary to compute information for sounds that are drowned out or otherwise not heard by a listener.

Brief Description of the Drawings

Figure 1 shows a high-level view block diagram of a system including mixed media rendering.

Figure 2 shows a block diagram of a terminal that is used in a system for mixed media rendering.

Figure 3 shows a graph structure that is used in a system for mixed media rendering.

Figure 4 shows a process flow diagram of a method including mixed media rendering.

Figure 5 is a block diagram of an exemplary sequential scene in a mixed media presentation that shows how a system for rendering mixed media minimizes the amount of computation required to render a scene.

Detailed Description of the Preferred Embodiment

In the description herein, a preferred embodiment of the invention is described, including preferred process steps and data structures. Those skilled in the art would realize, after perusal of this application, that embodiments of the invention might be implemented using a variety of other techniques not specifically described, without undue experimentation or further invention, and that such other techniques would be within the scope and spirit of the invention.

Lexicon

The following terms relate or refer to aspects of the invention or its embodiments. The general meaning of each of these terms is intended to be illustrative and in no way limiting.

- **Binary scene decoder** – as used herein, the term "binary scene decoder" includes a device for (1) creating a scene graph and presenter graph, and (2) computing how to display information to a viewer. The binary scene decoder can be included in a client side terminal.

- **Scene graph** – as used herein, the term "scene graph" is a tree structure that includes a set of nodes that are placeholders for information regarding the display of an object in a mixed media scene. An exemplary scene graph may include a placeholder for a rectangular area corresponding to the object and a set of placeholders for various properties associated with that object, such as color, transparency and texture. Taken by itself, the scene graph is static and acts as a container for information. Although displayed as a tree, the scene graph describes a data structure.

- **Presenter graph** – as used herein, the term "presenter graph" is a tree structure that includes data relating to the display of an object in a mixed media scene. An exemplary presenter graph may include information relating to various properties of a media object, such as color, transparency and texture, as well as the information needed to compute

any drawing operations relating to those properties. Although displayed as a tree, the presenter graph describes a data structure for processing information.

- **terminal** – as used herein, the term "terminal" includes a client device that is used to receive and decode one or more media streams for display to a user. The terminal may include a computing device coupled to a network, a television with a set-top box coupled to a network or other devices.

- **MPEG-4** — as used herein, the term "MPEG-4" refers to a technique for encoding different forms of audio-visual data, called audio-visual objects. MPEG-4 involves defining these objects, describing their spatial and temporal relationship to each other and encoding them.

- **occlude** — as used herein, the term "occlude" refers to the movement of one object included in an MPEG-4 scene in relation to another object such that one of the objects is completely or partially obscured.

- **scene** — as used herein, the term "scene" refers to a set of objects and other elements (for example, sprites) that are present at any one point in time during a multi-media display.

- **translucent** or **transparent** — as used here, the terms "translucent" and "transparent" refer to properties of an object in a scene. If an object is translucent or transparent, it is necessary to render both the object and the objects behind it.

The scope and spirit of the invention is not limited to any of these definitions, or to specific examples mentioned therein, but is intended to include the most general concepts embodied by these and other terms.

System Elements

Figure 1 shows a high level view block diagram of a system including mixed media rendering.

A system 100 includes an originating server 110, a terminal 120, a communication link 130 and a media stream 140.

The originating server 110 includes a processor, a memory and sufficient server software so as to transmit a media stream 140 (for example, an MPEG-4 presentation) to a terminal 120. In different embodiments, the originating server 110 can multicast the media stream 140 to many terminals 120 or may unicast the media stream 140 to a single terminal 120. In one embodiment, all of the different aspects of a presentation are sent from a single originating server 110 to a terminal 120. In other embodiments, several originating servers 110 may be used to transmit different forms of media or different content to the terminal 120 where the media is combined to form a single presentation or scene in a presentation.

In another embodiment, the media stream 140 originates from a local file on the client side. In such embodiments, there is no need for an originating server 110.

The terminal 120 (shown in greater detail in figure 2) is under the control of a user 122. The terminal 120 preferably includes a buffer for storing media and sufficient circuitry or software for presenting the media stream to a user 122. The terminal 120 receives the media stream 140 from an originating server 110, buffers and decodes that stream 140, and presents it to the user 122. In one embodiment, the terminal 120 may receive different media streams 140 from different originating servers 110. The different media streams 140 are integrated at the terminal 120 so as to comprise a single presentation for a viewer 122. In some embodiments, the media is stored locally on the client side. In such embodiments, the terminal 120 does not receive the media stream 140 from the originating server 110, but rather from a local source, such as a file stored on a disk or in the memory on the user's computer.

Various embodiments of the terminal 120 include a computer and monitor, a television and set-top box, and other similar devices used to process information and present it to a user.

The communication link 130 can include a computer network, such as an Internet, intranet, extranet or a virtual private network. In other embodiments, the communication link 130 can include a direct communication line, a switched network such as a telephone network, a wireless network, a form of packet transmission or some combination thereof. All variations of communication links noted herein are also known in the art of computer communication. In one embodiment, the originating server 110 and the terminal 120 are coupled by the communication link 130.

The media stream 140 may include various forms of mixed media, preferably MPEG-4. The scenes in the media stream 140 that are sent from the originating server 110 or local memory are encoded in a binary format that is subsequently decoded at the terminal 120. Different types of media that may be included in a media stream 140 include audio, video, and vector graphics.

Figure 2 shows a block diagram of a terminal that is used in a system including mixed media rendering.

The terminal 120 includes input port 202, a buffer 204, a binary scene decoder 206, a rasterizer 208, an output port 210. In one embodiment, the terminal 120 is coupled to a presentation element 212. In various embodiments, the elements of the terminal 120 and presentation element 212 may be included in a set top box and television set, a computer and computer monitor or other devices.

The input port 202 includes a port for receiving a digital media stream 140. The media stream 140 is received at the input port 202 and buffered in the buffer 204 until such time that it is decoded by the binary scene decoder 206.

The binary scene decoder 206 includes a processor and sufficient memory for storing instructions relating to generating a scene graph and a presenter graph and sending information relating to those graphs to the rasterizer 208. Both the scene graph and presenter graph are described in detail in figure 3. However, in brief, the scene graph is organized as a tree of placeholders for each object in a scene. The presenter graph includes information relating to the properties associated with the scene graph and how they relate to each other.

The binary scene decoder 206 also includes a set of directions for rendering information described by the presenter graph using the smallest number of drawing operations.

The rasterizer 208 includes a processor and a computer program for generating a bit map and a set of pixels that are responsive to information from the binary scene decoder 206. In one embodiment, the drawing process implemented by the rasterizer 208 is optimized for rendering MPEG4 data; however, this process can be optimized for rendering other information such as a presentations that are encoded using MPEG-1, MPEG-2, H.261, H.263, ITU-T or other similar encoding techniques. Upon generating a set of pixels, the rasterizer 208 sends the set of pixels to the output port 210 for presentation to a user 122 at a presentation element 212.

The presentation element 212 includes an element such as a television screen, computer monitor or other viewing platform for presenting the set of pixels generated by the rasterizer 208 to a user 122.

Figure 3 shows graph structures that are used in a system for mixed media rendering. Although presented herein as a tree, the graph structures represent data structures for processing instructions.

As described above, the binary scene decoder 206 transforms each object in a scene in the data stream 140 into a pair of graphs 300, including a scene graph 310, a presenter graph 330 and a set of links 350 between the scene graph 310 and the presenter graph 330.

The scene graph 310 includes a root node 315 and a set of dependent nodes 320. The root node 315 describes a shape (such as a rectangle or square) that is associated with the object. The relationship between this shape and the object is further described in figure 4. The dependent nodes 320 each describe a property associated with the object. These properties can include rotation, transparency, color, volume, texture, size and other similar characteristics. One of the root nodes 315 may couple all of the dependent nodes 320 associated with a scene. Such root nodes 315 are also known as join nodes.

The information stored in the scene graph 310 provides a snapshot of the current state of a scene, including a set of values related to the particular node (for example, values associated with transparency or values associated with color). However, neither the root node 315 nor the dependent nodes 320 include any intelligence regarding the rendering of the objects. The information included in the scene graph 310 reflects a static representation of the scene because it does not include any intelligence relating to how those values may change over time or in response to interactions with a user 122.

The presenter graph 330 includes a root node 335 and a set of dependent nodes 340 arranged as a DAG (direct acyclic graph). The root node 335 describes a shape (such as a rectangle or square) that is associated with the object. Similar to the root node 315, a root node 335 may couple all of the dependent nodes 340 associated with a scene. Such root nodes 335 are also known as join nodes.

The dependent nodes 340 each correspond to a property associated with the object. These properties can include rotation, transparency, color, volume, texture, size and other similar characteristics. However, unlike the scene graph 310 which includes a set of values related to each property, the root node 335 and dependent nodes 340 in a presenter graph 330 include additional intelligence for rendering each property associated with the node. In addition to knowing how to render that property, each dependent node 340 in the presenter graph 330 also knows how to manage interactions from a user 122 (for example, displaying a flash animation in response to user input). Lastly, the presenter graph 330 also includes information regarding which objects have been modified and instructions for generating the minimum number of drawing operations for the rasterizer 208.

Interactivity, animation and rendering are efficiently achieved by traversing the presenter graph 330 to obtain intelligence with respect to a value associated with the scene graph 310. As described in further detail in figure 4, two traversals of the presenter graph are required. The first traversal is directed at determining which objects need to be updated and the order in which the objects should be updated. The second traversal updates those objects that need to be updated. Objects or parts of objects that should not be visible to the user are not updated.

A set of links 350 between nodes in the scene graph 310 and the nodes in the presenter graph 330 create an association between various nodes such that there is a one-to-one relationship between individual nodes in the scene graph 310 and individual nodes in the presenter graph 330. Thus, when a value associated with a node in the scene graph 310 is modified, information regarding the change is sent over the link 350 to the corresponding node field in the presenter graph 330 so that the corresponding node is modified as well. For example, if a link 350 couples a node 315 is in the scene graph 310 with a node 340 in the presenter graph 330 and there is a change in the color value associated with node 315, then there will be a corresponding change regarding the instructions for rendering that color in node 340. Similar modifications can be made for other characteristics associated with objects such as rotation, transparency, color, volume, texture and others.

Method of Operation

Figure 4 shows a process flow diagram of a method including mixed media rendering.

A method 400 includes a set of flow points and a set of steps. In one embodiment, the system 100 performs the method 400, although the method 400 can be performed by other systems. Although the method 400 is described serially, the steps of the method 400 can be performed by separate elements in conjunction or in parallel, whether asynchronously, in a pipelined manner, or otherwise. There is no particular requirement that the method 400 be performed in the same order in which this description lists the steps, except where so indicated.

At a flow point 410, the system 100 is ready to begin performing a method 400.

In a step 415, a mixed media stream 140 is received from the originating server 110 or the local memory by the imput port 202 on the terminal 120. This mixed media stream 140 preferably includes an MPEG-4 presentation, but may also include a presentation that is encoded using MPEG-1, MPEG-2, H.261, H.263,ITU-T or similar encoding techniques.

In a step 420, the media stream 140 is buffered in the buffer 204 until such time that the binary scene decoder 206 is ready to decode it.

In a step 425, the binary scene decoder 206 selects sequential scenes in the media stream 140 to encode. The values of the objects in the scene are identified and a scene graph 310 and corresponding presenter graph 330 are generated.

In a step 430, the presenter graph 330 is traversed. The process of traversing (also known as processing) the presenter graph 330 involves identifying nodes 335 and dependent nodes 340 that are in need of updating. For example, if the size of an object changes from one scene to another, then the size of a rectangle corresponding to that object will also change. Information included in the presenter graph 330 that pertains to that rectangle will need to be updated to reflect these changes. Generally, an geometric shape associated with an object needs to be updated if the object changed (for example, changed in size, color, rotation, volume or other p+roperties) or because the object has become are partially occluded, completely occluded, or exposed by another object. As the presenter graph 330 is traversed, a list is generated of the geometric shapes and the dependent nodes associated with those shapes that need to be updated.

The traversal of the presenter graph 430 is performed depth first or substantially depth first. In this context, a traversal is considered substantially depth first if more than half of the traversal occurs depth first. The order of rectangles on the list of rectangles that need to be updated reflects the order in which they will be updated. For example, a scene comprised of a background and three objects may be designated as 0 (for the background) and 01, 02 and 03. These objects correspond respectively to different rectangles in the nodes in the tree, designated 0, R1, R2 and R3. As the order of rectangles corresponds to their relative location in the presenter graph 330, R2 is located deeper in the tree than R1. Similarly, R3 is located deeper in the tree than R2. These locations correspond to the actual spatial location of the objects. If an object is associated with a node that is deeper in the tree, that object is presented on top of or in front of other objects associated with nodes that are not as deep in the tree. If an object is not as deep in the tree, it may not need rendering because it may be partially or totally occluded by another object.

In a step 440, the presenter graph 330 is traversed a second time. During this second traversal, the nodes that require updating are updated in the order on the list. This prevents objects from being unnecessarily redisplayed. If the object is completely obscured, it will not be updated. This second traversal is performed depth first or substantially depth first. A traversal is performed substantially depth first if more than half the traversal is performed depth first.

In a step 445, the instructions from the presenter graph 330 for updating objects are sent to the rasterizer 208.

In a step 450, the rasterizer 208 generates a bit map that corresponds to the updated information.

In a step 455, the updated bit map is sent to the presentation element 212 for presentation to a user 122.

The method 400 may be performed for every scene in a mixed media presentation so as to efficiently render the media stream 140 for display to a user 122.

Figure 5 is a block diagram of an exemplary sequential scene in a mixed media presentation that shows how a system for rendering mixed media minimizes the amount of computation required to render a scene.

The exemplary sequential scene 500 is comprised of a first scene 510 and a second scene 530. The second scene 530 occurs at a point in time just after the first scene 510. A mixed media presentation may include many scenes, which are comprised of many objects.

First scene 510 and second scene 530 both include a square 515, a small circle 520 and a large circle 525. The difference between the first scene 510 and the second scene 530 is that the small circle 520 has moved from the right of the square 515 to the left of the square 515. A rectangle 540 is used to define the area where this movement takes place. The rectangle 540 defines the region that needs to be updated.

A scene graph 310 is used to provide a placeholder for the values associated with the square 515, the small circle 520 and the large circle 525. A presenter graph 330 indicates the changes between scene 510 and scene 530. By traversing the presenter graph 330 twice as shown in figure 4, a list of the elements that need updating is generated and updates are performed for only those elements that need updating. For example, it is only necessary to update the area bounded by rectangle 540. It is not necessary to update the parts of the 515 that are not bounded by the rectangle 540 or the large circle 525.

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention. These variations would become clear to those skilled in the art after perusal of this application.

Claims

1. A method, including steps of

generating a first data graph, in response to a set of values associated with at least one object included in a data stream associated with a mixed media presentation;

generating a second data graph responsive to a set of changes in said values included in said first data graph;

processing a set of nodes included in said second data graph substantially depth first and generating a list of said changes responsive to an order in which said nodes were processed;

processing said set of nodes substantially depth first and updating a set of objects in response to said order of said list;

rendering said objects responsive to said order;

whereby said objects that are visible or audible to a user are rendered and said objects that have not been changed or are not visible or audible to said user are not rendered.

2. A method as in claim 1, wherein said first data graph includes values associated with at least one of the following aspects of said object: size, color, position, angular position, texture, transparency and shape.

3. A method as in claim 2, wherein said aspects are used to animate said object.

4. A method as in claim 1, wherein said second data graph includes information about rendering changes in said values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

5. A method as in claim 1, wherein said changes are responsive to input from a user.

6. A method as in claim 1, wherein the order in which said changes are described on said list is responsive to their relative depth in said second data graph.

7. A method as in claim 1, wherein said step of processing said first data graph is performed substantially depth first.

8. A method as in claim 1, wherein said step of processing said second data graph is performed substantially depth first.

9. A method as in claim 1, wherein said step of rendering said objects includes generating a bit map and a set of pixels.

10. A method as in claim 1, wherein said data stream includes at least one of the following: MPEG-1, MPEG-2, H.261, H.263, ITU-T or MPEG-4.

11. A method as in claim 1, wherein said step of rendering is responsive to the relative transparency of said objects.

12. An apparatus, including

a means for generating a first data graph including a set of nodes, in response to values associated with at least one object included in said mixed media scene;

a means for generating a second data graph, including a set of nodes, wherein said second data graph is responsive to one or more changes in said first data graph;

a means for processing said second data graph so as to generate a list of said one or more changes, wherein the order of one or more changes on said list is responsive to a location in said second data graph;

a means for processing said second data graph and updating objects in response to said list;

a means for rendering said objects in an order responsive to the order in which they were updated such that said objects that are visible or audible to a user are rendered and said objects that are not visible or audible to said user are not rendered.

13. An apparatus as in claim 12, wherein said means for generating a first data graph include a binary scene decoder.

14. An apparatus as in claim 12, wherein said means for generating a second data graph include a binary scene decoder.

15. An apparatus as in claim 12, wherein said means for processing said second data graph include a binary scene decoder.

16. An apparatus as in claim 12, wherein at least some of said nodes in said first scene graph are linked to a corresponding said node in said second data graph.

17. An apparatus as in claim 12, wherein said first data graph includes a scene graph.

18. An apparatus as in claim 12, wherein first data graph includes values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

19. An apparatus as in claim 12, wherein said second data graph includes a direct acylic graph.

20. An apparatus as in claim 12, wherein said second data graph includes information about rendering changes in values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

21. An apparatus as in claim 12, wherein said changes are responsive to input from a user.

22. An apparatus as in claim 12, wherein said means for processing process said second data graph method substantially depth first.

23. A apparatus as in claim 12, wherein said means for rendering includes a rasterizer.

24. An apparatus, including

a computing device, having a memory;

a binary scene decoder stored in said memory, including a first data graph and a second data graph, wherein said first data graph having been generated by said binary scene decoder in response to a scene included in a mixed media presentation, and said second data graph having been generated in response to a set of changes in values associated with said first data graph; and

a rasterizer having an input from said second data graph, wherein said input is used to rendering objects that are visible or audible to a user and not render objects that are not visible or audible or which have already been rendered.

25. An apparatus as in claim 24, wherein said first data graph includes a set of nodes at least one of which is linked to a set of nodes included in said second data graph.

26. An apparatus as in claim 24, wherein said first data graph is a scene graph.

27. An apparatus as in claim 24, wherein first data graph includes values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

28. An apparatus as in claim 24, wherein said second data graph includes a direct acylic graph.

29. An apparatus as in claim 24, wherein said second data graph includes information about rendering changes in values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

30. An apparatus as in claim 24, wherein said changes are responsive to input from a user.

31. A memory storing information including instructions, wherein the instructions are executable by a processor, the instructions including method, including instructions of

generating a first data graph, in response to a set of values associated with at least one object included in a data stream associated with a mixed media presentation;

generating a second data graph responsive to a set of changes in said values included in said first data graph;

processing a set of nodes included in said second data graph substantially depth first and generating a list of changes responsive to said nodes

processing said set of nodes substantially depth first and updating a set of objects in response to an ordering of said list;

rendering said objects responsive to the order in which they were updated

whereby said objects that are visible or audible to a user are rendered and said objects that are not visible or audible to said user are not rendered.

32. A memory as in claim 31, wherein said first data graph includes values associated with at least one of the following aspects of said object: size, color, position, angular position, texture, transparency and shape.

33. A memory as in claim 32, wherein said aspects are used to animate said object.

34. A method as in claim 31, wherein said second data graph includes information about rendering changes in said values associated with at least one of the following aspects of said object: size, color, position, rotation, texture, transparency, and shape.

35. A memory as in claim 31, wherein said changes are responsive to input from a user.

36. A memory as in claim 31, wherein the order in which said changes are described on said list is responsive to their relative depth in said second data graph.

37. A memory as in claim 31, wherein said instruction for processing said first data graph is performed substantially depth first.

38. A memory as in claim 31, wherein said instruction for processing said second data graph is performed substantially depth first.

39. A memory as in claim 31, wherein said instruction for rendering said objects includes generating a bit map and a set of pixels.

40. A memory as in claim 31, wherein said data stream includes at least one of the following: MPEG-1, MPEG-2, H.261, H.263, ITU-T or MPEG-4.

41. A memory as in claim 31, wherein said instruction for rendering is responsive to the relative transparency of said objects.

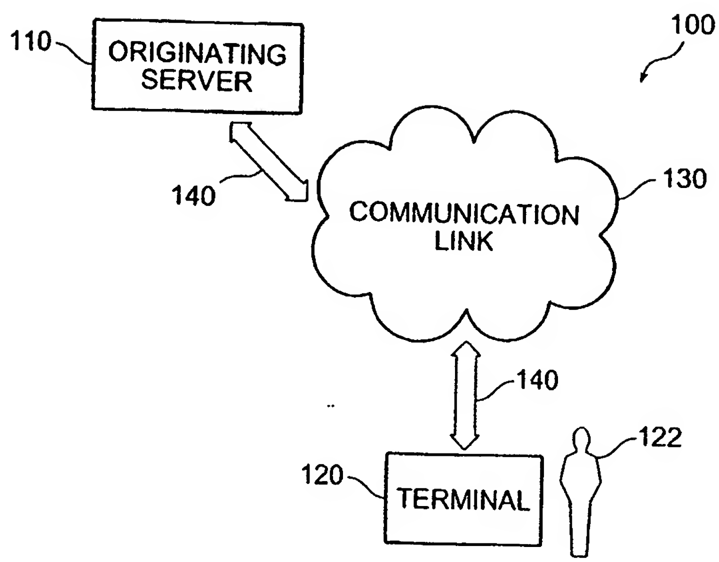

110
ORIGINATING SERVER
100
140
COMMUNICATION LINK
130
140
122
120
TERMINAL

FIG. 1

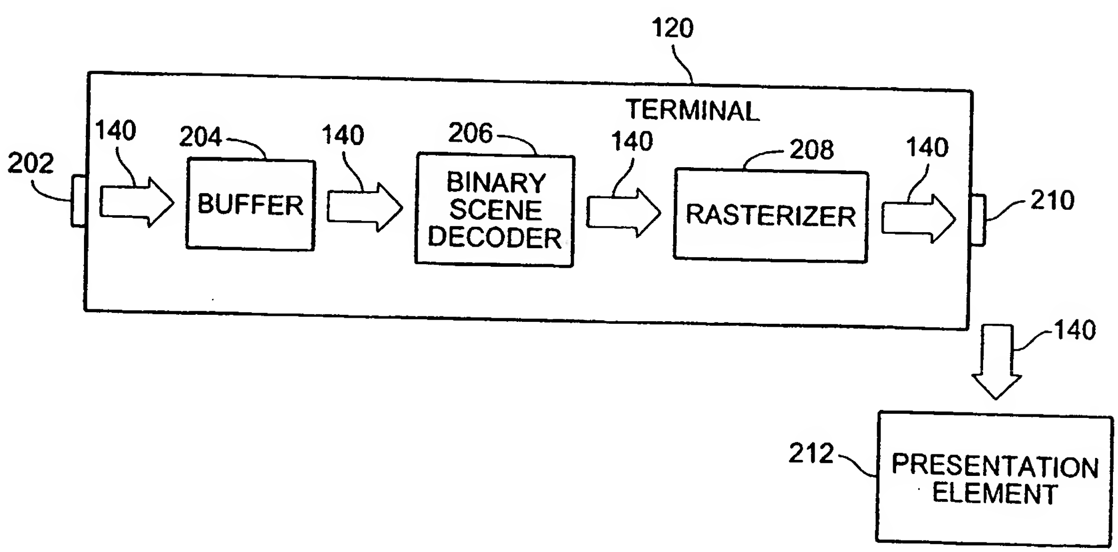

120
TERMINAL
202
140
204
BUFFER
140
206
BINARY SCENE DECODER
140
208
RASTERIZER
140
210
140
212
PRESENTATION ELEMENT

FIG. 2

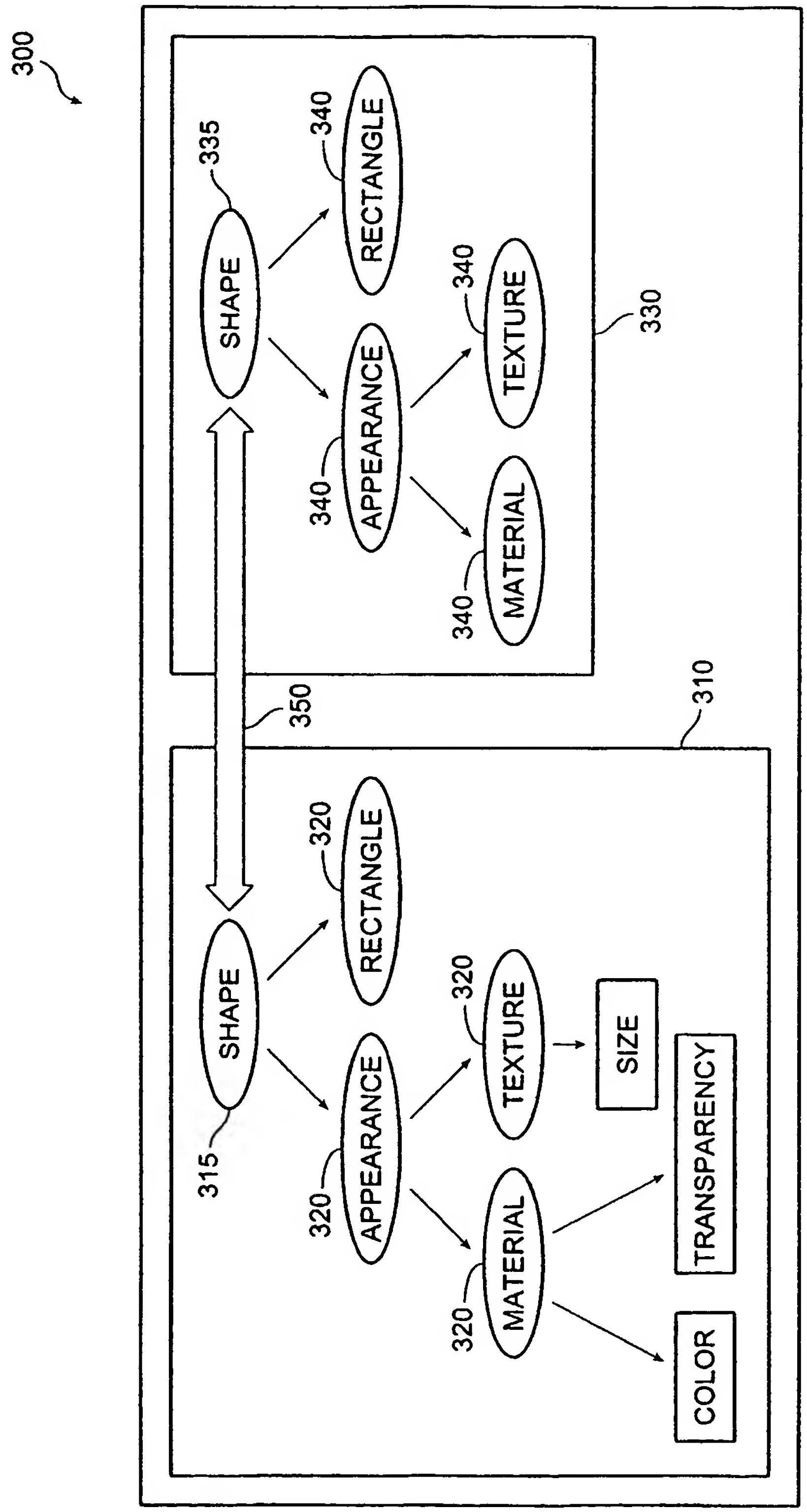

300
SHAPE
335
RECTANGLE
340
APPEARANCE
340
TEXTURE
340
MATERIAL
340
330
350
310
SHAPE
315
RECTANGLE
320
APPEARANCE
320
TEXTURE
320
MATERIAL
320
SIZE
TRANSPARENCY
COLOR

FIG. 3

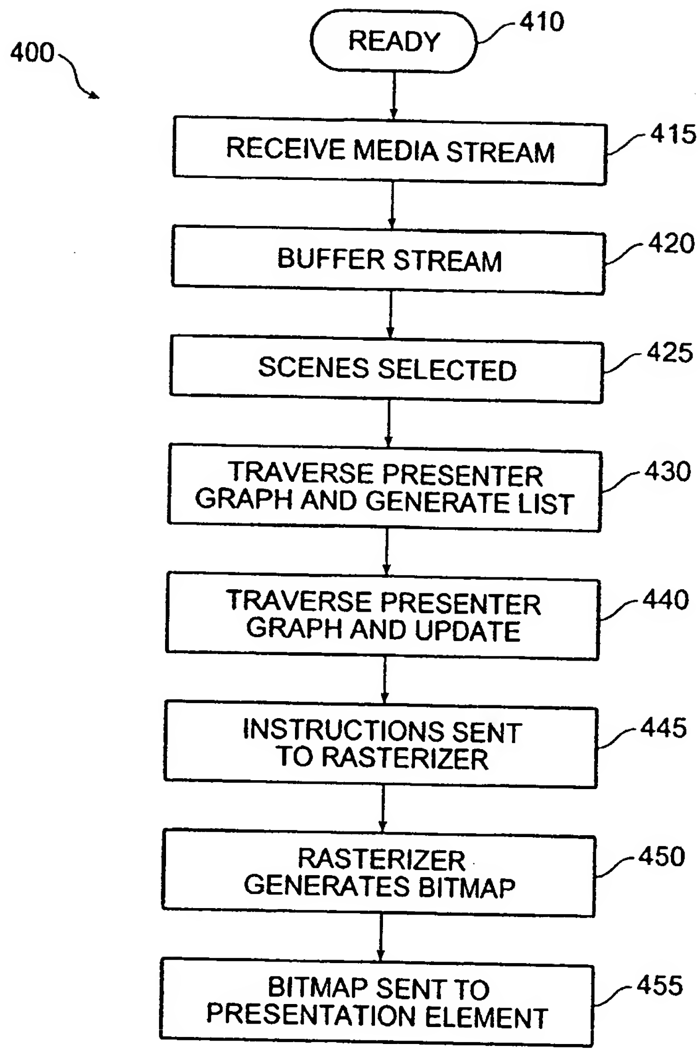

400
READY
410
RECEIVE MEDIA STREAM
415
BUFFER STREAM
420
SCENES SELECTED
425
TRAVERSE PRESENTER GRAPH AND GENERATE LIST
430
TRAVERSE PRESENTER GRAPH AND UPDATE
440
INSTRUCTIONS SENT TO RASTERIZER
445
RASTERIZER GENERATES BITMAP
450
BITMAP SENT TO PRESENTATION ELEMENT
455

FIG. 4

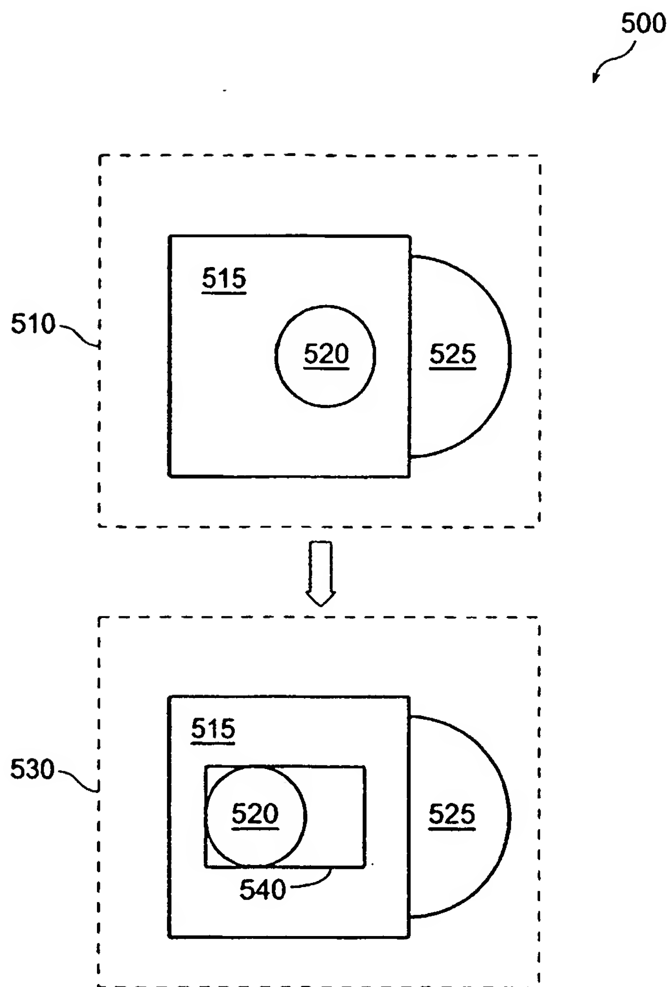


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/17088

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06T 11/20
US CL : 345/440

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 345/419, 421, 427, 428, 440, 581, 7 13, 716

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,608,899 A (LI et al.) 04 March 1997, entire document.	1-41
A	US 5,734,888 A (LI et al.) 31 March 1998, entire document	1-41
A,T	US 6,606,306 B1 (LIN et al.) 12 August 2003, entire document.	1-41
A,P	US 2002/0188959 A1 (PIOTROWSKI) 12 December 2002, entire document.	1-41
A,P	US 6,470,383 B1 (LESHEM et al.) 22 October 2002, ALL	1-41
A,P	US 6,463,431 B1 (SCHMITT) 08 October 2002, ALL	1-41

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
26 August 2003 (26.08.2003)

Date of mailing of the international search report
24 SEP 2003

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
Facsimile No. (703)305-3230

Authorized officer
Mark Zimmerman
Kugenia Logan
Telephone No. 703 306-0377

INTERNATIONAL SEARCH REPORT

PCT/US03/17088

Continuation of B. FIELDS SEARCHED Item 3:

USPT, PGPB (((345/419 |345/421 |345/427 |345/428 |345/440 |345/581 |345/713 |345/716)!.CCLS.)) and data same graph and nodes and size and color and position and hierarch$7 and list and map$6

USPT, PGPB (((345/$)!.CCLS.)) and data same graph and nodes and size and color and position and hierarch$7 and list and map$6

data same graph and nodes and size and color and position and hierarch$7 and list and map$6

data same graph and nodes and size and color and position and hierarch$7 and list

data same graph and nodes and size and color and position and hierarch$7

data same graph and nodes and size and color and position

USPT, PGPB (mix adj media) same stream$4

USPT, PGPB (((345/$)!.CCLS.)) and (mix adj media)

USPT, PGPB (mix adj media) same stream$4

USPT, PGPB (((345/$)!.CCLS.)) and (mix adj media)

USPT, PGPB ((345/$)!.CCLS.) and (mix adj media) same stream$4

USPT, PGPB (mix adj media) and stream$4